**Unit-3**

**Data Mining**:

Data mining in general terms means mining or digging deep into data that is in different forms to gain patterns, and to gain knowledge on that pattern. In the process of data mining, large data sets are first sorted, then patterns are identified and relationships are established to perform data analysis and solve problems.

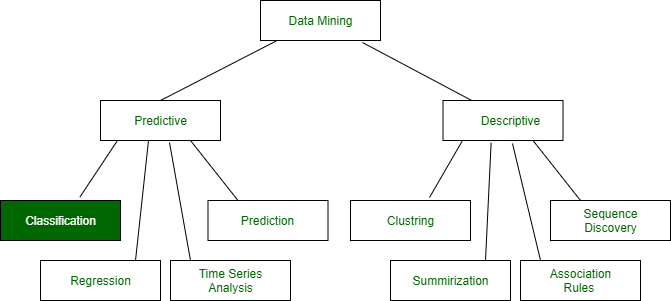
Classification is a task in data mining that involves assigning a class label to each instance in a dataset based on its features. The goal of classification is to build a model that accurately predicts the class labels of new instances based on their features.

There are two main types of classification: binary classification and multi-class classification. Binary classification involves classifying instances into two classes, such as “spam” or “not spam”, while multi-class classification involves classifying instances into more than two classes.

**The process of building a classification model typically involves the following steps:**

1. Data preparation: This step involves cleaning and pre-processing the data, such as removing missing values and transforming the data into a format that can be used by the classification algorithm.
2. Model selection: This step involves choosing an appropriate classification algorithm based on the characteristics of the data and the desired outcome. Common algorithms include decision trees, k-nearest neighbors, and support vector machines.
3. Model training: This step involves using the training data to train the classification algorithm and build the model. The model is trained by adjusting its parameters to minimize the difference between the predicted class labels and the actual class labels.
4. Model evaluation: This step involves evaluating the performance of the classification model on a test dataset that is separate from the training data. This can be done by calculating metrics such as accuracy, precision, recall, and F1-score.
5. Model deployment: This step involves deploying the classification model in a production environment, where it can be used to make predictions on new instances.

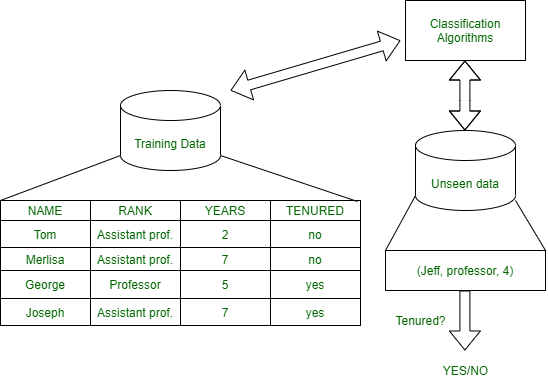
Classification is a widely used technique in data mining and is applied in a variety of domains, such as email filtering, sentiment analysis, and medical diagnosis.



**Classification**: It is a data analysis task, i.e. the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

**Example**: Before starting any project, we need to check its feasibility. In this case, a classifier is required to predict class labels such as ‘Safe’ and ‘Risky’ for adopting the Project and to further approve it. It is a two-step process such as:

1. **Learning Step (Training Phase)**: Construction of Classification Model   
   Different Algorithms are used to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.
2. **Classification Step**: Model used to predict class labels and testing the constructed model on test data and hence estimate the accuracy of the classification rules.



*Test data are used to estimate the accuracy of the classification rule*

**Training and Testing:**

Suppose there is a person who is sitting under a fan and the fan starts falling on him, he should get aside in order not to get hurt. So, this is his training part to move away. While Testing if the person sees any heavy object coming towards him or falling on him and moves aside then the system is tested positively and if the person does not move aside then the system is negatively tested.   
The same is the case with the data, it should be trained in order to get the accurate and best results.

There are certain data types associated with data mining that actually tells us the format of the file (whether it is in text format or in numerical format).

Attributes – Represents different features of an object. Different types of attributes are:

1. **Binary**: Possesses only two values i.e. True or False   
   Example: Suppose there is a survey evaluating some products. We need to check whether it’s useful or not. So, the Customer has to answer it in Yes or No.   
   Product usefulness: Yes / No
   * **Symmetric**: Both values are equally important in all aspects
   * **Asymmetric**: When both the values may not be important.
2. **Nominal**: When more than two outcomes are possible. It is in Alphabet form rather than being in Integer form.   
   **Example**: One needs to choose some material but of different colors. So, the color might be Yellow, Green, Black, Red.   
   Different Colors: Red, Green, Black, Yellow
   * **Ordinal**: Values that must have some meaningful order.   
     Example: Suppose there are grade sheets of few students which might contain different grades as per their performance such as A, B, C, D   
     Grades: A, B, C, D
   * **Continuous**: May have an infinite number of values, it is in float type   
     Example: Measuring the weight of few Students in a sequence or orderly manner i.e. 50, 51, 52, 53   
     Weight: 50, 51, 52, 53
   * **Discrete**: Finite number of values.   
     Example: Marks of a Student in a few subjects: 65, 70, 75, 80, 90   
     Marks: 65, 70, 75, 80, 90

**Syntax:**

* Mathematical Notation: Classification is based on building a function taking input feature vector “X” and predicting its outcome “Y” (Qualitative response taking values in set C)
* Here Classifier (or model) is used which is a Supervised function, can be designed manually based on the expert’s knowledge. It has been constructed to predict class labels (Example: Label – “Yes” or “No” for the approval of some event).

Classifiers can be categorized into two major types: 

1. **Discriminative**: It is a very basic classifier and determines just one class for each row of data. It tries to model just by depending on the observed data, depends heavily on the quality of data rather than on distributions.   
   **Example**: Logistic Regression
2. **Generative**: It models the distribution of individual classes and tries to learn the model that generates the data behind the scenes by estimating assumptions and distributions of the model. Used to predict the unseen data.   
   **Example**: Naive Bayes Classifier   
   Detecting Spam emails by looking at the previous data. Suppose 100 emails and that too divided in 1:4 i.e. Class A: 25%(Spam emails) and Class B: 75%(Non-Spam emails). Now if a user wants to check that if an email contains the word cheap, then that may be termed as Spam.   
   It seems to be that in Class A(i.e. in 25% of data), 20 out of 25 emails are spam and rest not.   
   And in Class B(i.e. in 75% of data), 70 out of 75 emails are not spam and rest are spam.   
   So, if the email contains the word cheap, what is the probability of it being spam ?? (= 80%)

**Classifiers Of Machine Learning:**

1. Decision Trees
2. Bayesian Classifiers
3. Neural Networks
4. K-Nearest Neighbour
5. Support Vector Machines
6. Linear Regression
7. Logistic Regression

**Associated Tools and Languages:** Used to mine/ extract useful information from raw data.

* **Main Languages used**: R, SAS, Python, SQL
* **Major Tools used**: RapidMiner, Orange, KNIME, Spark, Weka
* **Libraries used**: Jupyter, NumPy, Matplotlib, Pandas, ScikitLearn, NLTK, TensorFlow, Seaborn, Basemap, etc.

**Real**–**Life Examples :**

* **Market Basket Analysis:**   
  It is a modeling technique that has been associated with frequent transactions of buying some combination of items.   
  **Example**: Amazon and many other Retailers use this technique. While viewing some products, certain suggestions for the commodities are shown that some people have bought in the past.
* **Weather Forecasting:**   
  Changing Patterns in weather conditions needs to be observed based on parameters such as temperature, humidity, wind direction. This keen observation also requires the use of previous records in order to predict it accurately.

**Advantages:**

* Mining Based Methods are cost-effective and efficient
* Helps in identifying criminal suspects
* Helps in predicting the risk of diseases
* Helps Banks and Financial Institutions to identify defaulters so that they may approve Cards, Loan, etc.

**Disadvantages:**  
Privacy: When the data is either are chances that a company may give some information about their customers to other vendors or use this information for their profit.   
Accuracy Problem: Selection of Accurate model must be there in order to get the best accuracy and result.

**APPLICATIONS:** 

* Marketing and Retailing
* Manufacturing
* Telecommunication Industry
* Intrusion Detection
* Education System
* Fraud Detection

**GIST OF DATA MINING :**

1. Choosing the correct classification method, like decision trees, Bayesian networks, or neural networks.
2. Need a sample of data, where all class values are known. Then the data will be divided into two parts, a training set, and a test set.

Now, the training set is given to a learning algorithm, which derives a classifier. Then the classifier is tested with the test set, where all class values are hidden.   
If the classifier classifies most cases in the test set correctly, it can be assumed that it works accurately also on the future data else it may be the wrong model chosen.

Data mining refers to extracting or mining knowledge from large amounts of data. In other words, data mining is the science, art, and technology of discovering large and complex bodies of data in order to discover useful patterns. Theoreticians and practitioners are continually seeking improved techniques to make the process more efficient, cost-effective, and accurate. Any situation can be analyzed in two ways in data mining:

* **Statistical Analysis:** In statistics, data is collected, analyzed, explored, and presented to identify patterns and trends. Alternatively, it is referred to as quantitative analysis.
* **Non-statistical Analysis:** This analysis provides generalized information and includes sound, still images, and moving images.

In statistics, there are two main categories:

* **Descriptive Statistics:** The purpose of descriptive statistics is to organize data and identify the main characteristics of that data. Graphs or numbers summarize the data. Average, Mode, SD(Standard Deviation), and Correlation are some of the commonly used descriptive statistical methods.
* **Inferential Statistics:** The process of drawing conclusions based on probability theory and generalizing the data. By analyzing sample statistics, you can infer parameters about populations and make models of relationships within data.

There are various statistical terms that one should be aware of while dealing with statistics. Some of these are:

* Population
* Sample
* Variable
* Quantitative Variable
* Qualitative Variable
* Discrete Variable
* Continuous Variable

Now, let’s start discussing statistical methods. This is the analysis of raw data using mathematical formulas, models, and techniques. Through the use of statistical methods, information is extracted from research data, and different ways are available to judge the robustness of research outputs.

As a matter of fact, today’s statistical methods used in the data mining field typically are derived from the vast statistical toolkit developed to answer problems arising in other fields. These techniques are taught in science curriculums. It is necessary to check and test several hypotheses. The hypotheses described above help us assess the validity of our data mining endeavor when attempting to infer any inferences from the data under study. When using more complex and sophisticated statistical estimators and tests, these issues become more pronounced.

For extracting knowledge from databases containing different types of observations, a variety of statistical methods are available in Data Mining and some of these are:

* Logistic regression analysis
* Correlation analysis
* Regression analysis
* Discriminate analysis
* Linear discriminant analysis (LDA)
* Classification
* Clustering
* Outlier detection
* Classification and regression trees,
* Correspondence analysis
* Nonparametric regression,
* Statistical pattern recognition,
* Categorical data analysis,
* Time-series methods for trends and periodicity
* Artificial neural networks

Now, let’s try to understand some of the important statistical methods which are used in data mining:

* **Linear Regression:** The linear regression method uses the best linear relationship between the independent and dependent variables to predict the target variable. In order to achieve the best fit, make sure that all the distances between the shape and the actual observations at each point are as small as possible. A good fit can be determined by determining that no other position would produce fewer errors given the shape chosen. Simple linear regression and multiple linear regression are the two major types of linear regression. By fitting a linear relationship to the independent variable, the simple linear regression predicts the dependent variable. Using multiple independent variables, multiple linear regression fits the best linear relationship with the dependent variable. For more details, you can refer [linear regression.](https://www.geeksforgeeks.org/linear-regression-python-implementation/#:~:text=Simple%20linear%20regression%20is%20an,or%20independent%20variable(x).)
* **Classification:** This is a method of data mining in which a collection of data is categorized so that a greater degree of accuracy can be predicted and analyzed. An effective way to analyze very large datasets is to classify them. Classification is one of several methods aimed at improving the efficiency of the analysis process. A Logistic Regression and a Discriminant Analysis stand out as two major classification techniques.
  + **Logistic Regression:** It can also be applied to machine learning applications and predictive analytics. In this approach, the dependent variable is either binary (binary regression) or multinomial (multinomial regression): either one of the two or a set of one, two, three, or four options. With a logistic regression equation, one can estimate probabilities regarding the relationship between the independent variable and the dependent variable. For understanding logistic regression analysis in detail, you can refer to logistic regression.
  + **Discriminant Analysis:** A Discriminant Analysis is a statistical method of analyzing data based on the measurements of categories or clusters and categorizing new observations into one or more populations that were identified a priori. The discriminant analysis models each response class independently then uses Bayes’s theorem to flip these projections around to estimate the likelihood of each response category given the value of X. These models can be either linear or quadratic.
    - **Linear Discriminant Analysis:**  According to **Linear Discriminant Analysis,** each observation is assigned a discriminant score to classify it into a response variable class. By combining the independent variables in a linear fashion, these scores can be obtained. Based on this model, observations are drawn from a Gaussian distribution, and the predictor variables are correlated across all k levels of the response variable, Y. and for further details [linear discriminant analysis](https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/)
    - **Quadratic Discriminant Analysis:**  An alternative approach is provided by Quadratic Discriminant Analysis. LDA and QDA both assume Gaussian distributions for the observations of the Y classes. Unlike LDA, QDA considers each class to have its own covariance matrix. As a result, the predictor variables have different variances across the k levels in Y.
  + **Correlation Analysis:** In statistical terms, correlation analysis captures the relationship between variables in a pair. The value of such variables is usually stored in a column or rows of a database table and represents a property of an object.
  + **Regression Analysis:** Based on a set of numeric data, regression is a data mining method that predicts a range of numerical values (also known as continuous values). You could, for instance, use regression to predict the cost of goods and services based on other variables. A regression model is used across numerous industries for forecasting financial data, modeling environmental conditions, and analyzing trends.

The first step in creating good statistics is having good data that was derived with an aim in mind. There are two main types of data: an input (independent or predictor) variable, which we control or are able to measure, and an output (dependent or response) variable which is observed. A few will be quantitative measurements, but others may be qualitative or categorical variables (called factors).

# Data Mining Algorithms

Data Mining Algorithms are a particular category of algorithms useful for analyzing data and developing data models to identify meaningful patterns. These are part of machine learning algorithms. These algorithms are implemented through various programming like R language, Python, and data mining tools to derive the optimized data models. Some of the popular data mining algorithms are C4.5 for decision trees, K-means for cluster data analysis, [**Naive Bayes Algorithm**](https://www.educba.com/naive-bayes-algorithm/), Support Vector Mechanism Algorithms, The Apriori algorithm for time series data mining. These algorithms are part of data analytics implementation for business. These algorithms are based upon statistical and mathematical formulas which applied to the data set.

#### Algorithm

Some constructs are used by classifiers which are tools in data mining. These systems take inputs from a collection of cases where each case belongs to one of the small numbers of classes and are described by its values for a fixed set of attributes. The output classifier can accurately predict the level to which it belongs. It uses decision trees where the first initial tree is acquired by using a divide and conquer algorithm.

Suppose S is a class and the tree is leaf labelled with the most frequent type in S. Choosing a test based on a single attribute with two or more outcomes than making this test as root one branch for each work of the test can be used. The partitions correspond to subsets S1, S2, etc., which are outcomes for each case. C4.5 allows for multiple products. C4.5 has introduced an alternative formula in thorny decision trees, which consists of a list of rules, where these rules are grouped for each class. To classify the case, the first class whose conditions are satisfied is named as the first one. If the patient meets no power, then it is assigned a default class. The C4.5 rulesets are formed from the initial decision tree. C4.5 enhances the scalability by multi-threading.

#### 2. The k-means Algorithm

This algorithm is a simple method of partitioning a given data set into the user-specified number of clusters. This algorithm works on d-dimensional vectors, D={xi | i= 1, … N} where i is the data point. To get these initial data seeds, the data has to be sampled at random. This sets the solution of clustering a small subset of data, the global mean of data k times. This algorithm can be paired with another algorithm to describe non-convex clusters. It creates k groups from the given set of objects. It explores the entire data set with its cluster analysis. It is simple and faster than other algorithms when it is used with different algorithms. This algorithm is mostly classified as semi-supervised. Along with specifying the number of clusters, it also keeps learning without any information. It observes the group and learns.

#### 3. Naive Bayes Algorithm

This algorithm is [**based on Bayes theorem**](https://www.educba.com/bayes-theorem/). This algorithm is mainly used when the dimensionality of inputs is high. This classifier can easily calculate the next possible output. New raw data can be added during the runtime, and it provides a better probabilistic classifier. Each class has a known set of vectors that aim to create a rule that allows the objects to be assigned to classes in the future. The vectors of variables describe the future things. This is one of the most comfortable algorithms as it is easy to construct and does not have any complicated parameter estimation schemas. It can be easily applied to massive data sets as well. It does not need any elaborate iterative parameter estimation schemes, and hence unskilled users can understand why the classifications are made.

#### 4. Support Vector Machines Algorithm

If a user wants robust and accurate methods, then Support Vector machines algorithm must be tried. SVMs are mainly used for learning classification, regression or ranking function. It is formed based on structural risk minimization and statistical learning theory. The decision boundaries must be identified, which is known as a hyperplane. It helps in the optimal separation of classes. The main job of SVM is to identify the maximizing the margin between two types. The margin is defined as the amount of space between two types. A hyperplane function is like an equation for the line, y= MX + b. SVM can be extended to perform numerical calculations as well. SVM makes use of kernel so that it operates well in higher dimensions. This is a supervised algorithm, and the data set is used first to let SVM know about all the classes. Once this is done then, SVM can be capable of classifying this new data.

#### 5. The Apriori Algorithm

The Apriori algorithm is widely used to find the frequent itemsets from a transaction data set and derive association rules. To find frequent itemsets is not difficult because of its combinatorial explosion. Once we get the frequent itemsets, it is clear to generate association rules for larger or equal specified minimum confidence. Apriori is an algorithm which helps in finding routine data sets by making use of candidate generation. It assumes that the item set or the items present are sorted in lexicographic order. After the introduction of Apriori data mining research has been specifically boosted. It is simple and easy to implement. The basic approach of this algorithm is as below:

* **Join**: The whole database is used for the hoe frequent 1 item sets.
* **Prune**: This item set must satisfy the support and confidence to move to the next round for the 2 item sets.
* **Repeat**: Until the pre-defined size is not reached till, then this is repeated for each itemset level

# based algorithms in data mining

The algorithms are used to measure the distance between each text and to calculate the score.



Distance measures play an important role in machine learning

They provide the foundations for many popular and effective machine learning algorithms like KNN (K-Nearest Neighbours) for supervised learning and K-Means clustering for unsupervised learning.

Different distance measures must be chosen and used depending on the types of data, As such, it is important to know how to implement and calculate a range of different popular distance measures and the intuitions for the resulting scores.

In this blog, we’ll discover distance measures in machine learning.

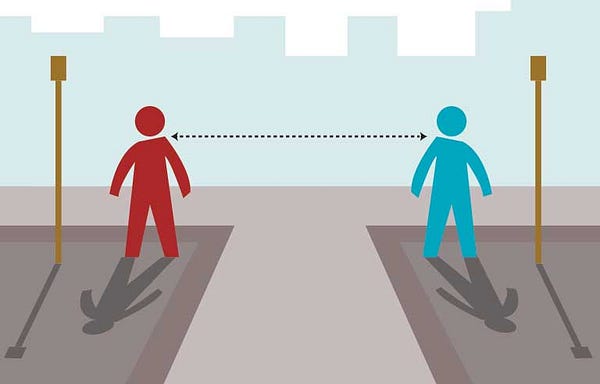
## **Overview:**

1. Role of Distance Measures
2. Hamming Distance
3. Euclidean Distance
4. Manhattan Distance (Taxiable or City Block)
5. Minkowski Distance
6. Mahalanobis Distance
7. Cosine Similarity

# **Role of Distance Measures**

Distance measures play an important role in machine learning

A distance measure is an objective score that summarizes the relative difference between two objects in a problem domain.

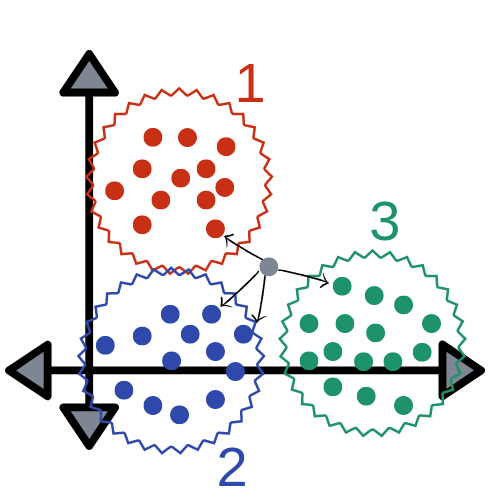


Most commonly, the two objects are rows of data that describes a subject (such as a person, car, or house), or an event (such as purchases, a claim, or a diagnosis)

Perhaps, the most likely way we can encounter distance measures is when we are using a specific machine learning algorithm that uses distance measures at its core. The most famous algorithm is KNN — [K-Nearest Neighbours Algorithm]

# **KNN**

A Classification or Regression prediction is made for new examples by calculating the distance between the new and all existing example sets in the training datasets.



The K examples in the training dataset with the smallest distance are then selected and a prediction is made by averaging the outcome(mode of the class label or mean of the real value for regression)

KNN belongs to a broader field of algorithms called case-based or instance-based learning, most of which uses distance measures in a similar manner. Another popular instance-based algorithm that uses distance measures is the learning vector quantization or LVQ, the algorithm that may also be considered a type of neural network.

Next, We have the Self-Organizing Map algorithm, or SOM, which is an algorithm that also uses distance measures and can be used for supervised and unsupervised learning algorithms that use distance measures at its core is the K-means clustering algorithm.

In Instance-Based Learning, the training examples are stored verbatim and a distance function is used to determine which member of the training set is closest to an unknown test instance. Once the nearest training instance has been located its class is predicted for the test instance.

A few of the More popular machine learning algorithms that use distance measures at their core is

1. K-Nearest Neighbors (KNN)
2. Learning Vector Quantization (LVQ)
3. Self-Organizing Map (SOM)
4. K-Means Clustering

There are many kernel-based methods that may also be considered distance-based algorithms.

Perhaps the most widely know kernel method is the Support Vector Machine algorithm (SVM)

*When Calculating the distance between two examples or rows of data, it is possible that different data types are used for different columns of the example set.*

The Example set might have real values, boolean, categorical, and ordinal values.

*Different distance measures may be required for each that are summed together into a single distance score.*

Numerical values may have different scales. this can greatly impact the calculation of distance measure and it is often a good practice to normalize or standardize numerical values prior to calculating the distance measure.

Numerical error in regression problems may also be considered a distance. For example error between the expected value and the predicted value is a one-dimensional distance measure that can be summed or averaged over all examples in a test set to give a total distance between the expected and predicted outcomes in the dataset.

The calculation of the errors, such as the mean squared error or mean absolute error, may resemble a standard distance measure.

As we can see, distance measures play an important role in machine learning,

the most commonly used distance measures in machine learning are

1. Hamming Distance
2. Euclidean Distance
3. Manhattan Distance
4. Minkowski Distance
5. Mahalanobis

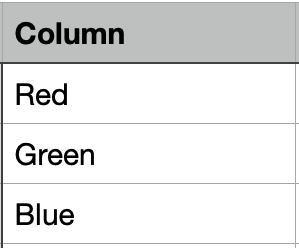
The most important is to know how to calculate each of these distance measures when implementing the algorithms from scratch and the intuition for what is being calculated when using algorithms that make use of these distance measures.

# **HAMMING DISTANCE**

Hamming distance calculates the distance between two binary vectors, also referred to as binary strings or bitstrings

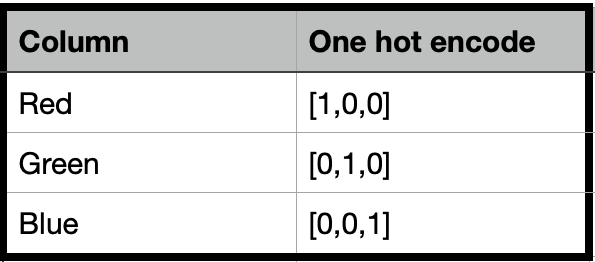
We are most likely going to encounter binary strings when we do One-Hot Encode categorical columns of data.

For example,



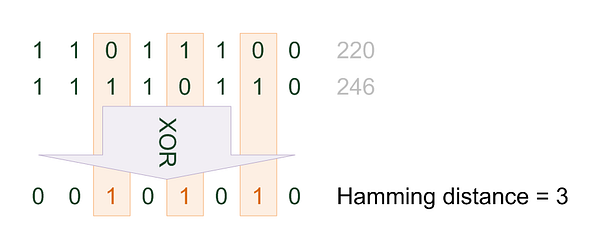
Example Set

After One Hot Encoding



An example set with one-hot encodes

the distance between red and green could be calculated as the sum or the average number of bit differences between the two bitstrings. This is Hamming distance.



For a One-hot encoded string, it might make more sense to summarize the sum of the bit difference between the strings, which will always be a 0 or 1.

* Hamming Distance = sum for i to N abs(v1[i] — v2[i])

For bitstrings that may have many 1 bits, it is more common to calculate the average number of bit differences to give a hamming distance score between 0(identical) and 1 (all different).

* Hamming Distance = (sum for i to N abs(v1[i] — v2[i]))/N

We can demonstrate this with an example of calculating the Hamming Distance between two bitstrings, listed below.

# calculating hamming distance between bit string  
# calculate hamming distance  
def hamming\_distance(a,b):  
 return sum(abs(e1-e2) for e1, e2 in zip(a,b)) / len(a)# define data  
row1 = [0,0,0,0,0,1]  
row2 = [0,0,0,0,1,0]# calculate distance  
dist = hamming\_distance(row1, row2)  
print(dist)

We can see that there are two differences between the strings, or 2 out of 6 bit different, which averaged (2/6) is about 1/3 or 0.33.

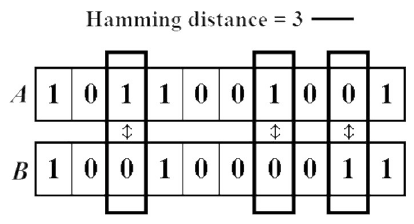
0.33333333333333

we can also perform the same calculation using hamming() function from SciPy.

# calculating hamming distance between bit strings  
from scipy.spatial.distance import hamming# define data  
row1 = [0, 0, 0, 0, 0, 1]  
row2 = [0, 0, 0, 0, 1, 0]# calculate distance  
dist = hamming(row1, row2)  
print(dist)

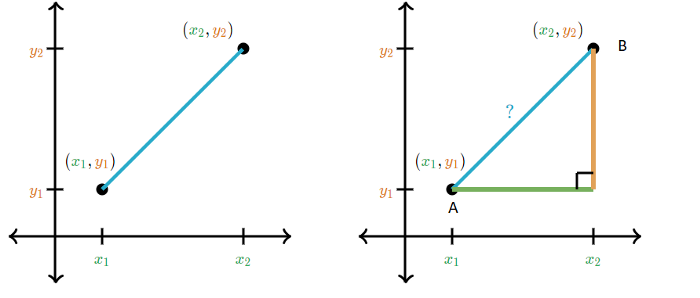
Here we can confirm the example we get the same results, confirming our manual implementation

0.33333333333333



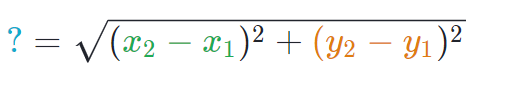
# **Euclidean Distance**

Euclidean distance calculates the distance between two real-valued vectors.



Taken from Khan Academy’s Distance Formula Tutorial

In order to calculate the distance between data points, the A and B Pythagorean theorem considers the length of the x and y-axis.



You are most likely to use Euclidean distance when calculating the distance between two rows of data that have numerical values, such a floating-point or integer values.

If columns have values with differing scales, it’s common to normalize or standardize the numerical values across all columns prior to calculating the euclidean distance. Otherwise, columns that have large values will dominate the distance measure.

Euclidean distance is calculated as the square root of the sum of the squared differences between the two vectors.

* EuclideanDistance = sqrt(sum for i to N (v1[i] — v2[i])²)

If the distance calculation is to be performed thousands or millions of times, it is common to remove the square root operation in an effort to speed up the calculation. The resulting scores will have the same relative proportions after this modification and can still be used effectively within a machine learning algorithm for finding the most similar examples.

* EuclideanDistance = sum for i to N (v1[i] — v2[i])²

This calculation is related to the L2 vector norm and is equivalent to the sum squared error and the root sum squared error if the square root is added.

We can demonstrate this with an example of calculating the Euclidean distance between two real-valued vectors, listed below.

# calculating euclidean distance between vectors  
from math import sqrt# calculate euclidean distance  
def euclidean\_distance(a, b):  
return sqrt(sum((e1-e2)\*\*2 for e1, e2 in zip(a,b)))# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance  
dist = euclidean\_distance(row1, row2)  
print(dist)

Running the example reports the Euclidean distance between the two vectors.

6.082762530298219

We can also perform the same calculation using the [euclidean() function](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.euclidean.html" \t "_blank) from SciPy. The complete example is listed below.

# calculating euclidean distance between vectors  
from scipy.spatial.distance import euclidean# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance  
dist = euclidean(row1, row2)  
print(dist)

Running the example, we can see we get the same result, confirming our manual implementation.

6.082762530298219

# Manhattan Distance (Taxicab or City Block Distance)

The [Manhattan distance](https://en.wikipedia.org/wiki/Taxicab_geometry), also called the Taxicab distance or the City Block distance, calculates the distance between two real-valued vectors.

It is perhaps more useful to vectors that describe objects on a uniform grid, like a chessboard or city blocks. The taxicab name for the measure refers to the intuition for what the measure calculates: the shortest path that a taxicab would take between city blocks (coordinates on the grid).

It might make sense to calculate Manhattan distance instead of Euclidean distance for two vectors in an integer feature space.

Manhattan distance is calculated as the sum of the absolute differences between the two vectors.

* ManhattanDistance = sum for i to N sum |v1[i] — v2[i]|

The Manhattan distance is related to the L1 vector norm and the sum absolute error and mean absolute error metric.

We can demonstrate this with an example of calculating the Manhattan distance between two integer vectors, listed below.

# calculating manhattan distance between vectors  
from math import sqrt# calculate manhattan distance  
def manhattan\_distance(a, b):  
return sum(abs(e1-e2) for e1, e2 in zip(a,b))# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance  
dist = manhattan\_distance(row1, row2)  
print(dist)

Running the example reports the Manhattan distance between the two vectors.

13

We can also perform the same calculation using the [cityblock() function](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cityblock.html" \t "_blank) from SciPy. The complete example is listed below.

# calculating manhattan distance between vectors  
from scipy.spatial.distance import cityblock# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance  
dist = cityblock(row1, row2)  
print(dist)

Running the example, we can see we get the same result, confirming our manual implementation.

13

# Minkowski Distance

[Minkowski distance](https://en.wikipedia.org/wiki/Minkowski_distance) calculates the distance between two real-valued vectors.

It is a generalization of the Euclidean and Manhattan distance measures and adds a parameter, called the “order” or “p“, that allows different distance measures to be calculated.

The Minkowski distance measure is calculated as follows:

* EuclideanDistance = (sum for i to N (abs(v1[i] — v2[i]))^p)^(1/p)

Where “p” is the order parameter.

When p is set to 1, the calculation is the same as the Manhattan distance. When p is set to 2, it is the same as the Euclidean distance.

* p=1: Manhattan distance.
* p=2: Euclidean distance.

Intermediate values provide a controlled balance between the two measures.

It is common to use Minkowski distance when implementing a machine learning algorithm that uses distance measures as it gives control over the type of distance measure used for real-valued vectors via a hyperparameter “p” that can be tuned.

We can demonstrate this calculation with an example of calculating the Minkowski distance between two real vectors, listed below.

# calculating minkowski distance between vectors  
from math import sqrt# calculate minkowski distance  
def minkowski\_distance(a, b, p):  
return sum(abs(e1-e2)\*\*p for e1, e2 in zip(a,b))\*\*(1/p)# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance (p=1)  
dist = minkowski\_distance(row1, row2, 1)  
print(dist)# calculate distance (p=2)  
dist = minkowski\_distance(row1, row2, 2)  
print(dist)

Running the example first calculates and prints the Minkowski distance with p set to 1 to give the Manhattan distance, then with p set to 2 to give the Euclidean distance, matching the values calculated on the same data from the previous sections.

13.0  
6.082762530298219

We can also perform the same calculation using the [minkowski\_distance() function](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.minkowski_distance.html" \t "_blank) from SciPy. The complete example is listed below.

# calculating minkowski distance between vectors  
from scipy.spatial import minkowski\_distance# define data  
row1 = [10, 20, 15, 10, 5]  
row2 = [12, 24, 18, 8, 7]# calculate distance (p=1)  
dist = minkowski\_distance(row1, row2, 1)  
print(dist)# calculate distance (p=2)  
dist = minkowski\_distance(row1, row2, 2)  
print(dist)

Running the example, we can see we get the same results, confirming our manual implementation.

13.0  
6.082762530298219

# Mahalanobis Distance

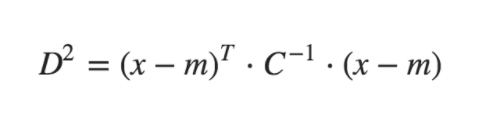
Mahalanobis distance is an effective multivariate distance metric that measures the distance between a point(vector) and a distribution. It is an extremely useful metric having, excellent applications in multivariate anomaly detection, classification on highly imbalanced datasets, and one-class classification.

It has excellent applications in multivariate anomaly detection, classification on highly imbalanced datasets and one-class classification and more untapped use cases.

Considering its extremely useful applications, this metric is seldom discussed or used in stats or ML workflows. This post explains the why and the when to use Mahalanobis distance and then explains the intuition and the math with useful applications.

Mahalanobis distance is the distance between a point and a distribution. And not between two distinct points. It is effectively a multivariate equivalent of the Euclidean distance

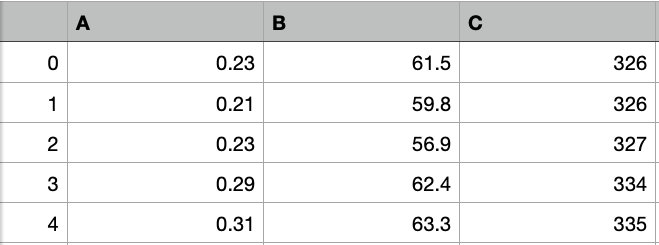
1. It transforms the columns into uncorrelated variables
2. Scale the columns to make their variance equal to 1
3. Finally, it calculates the Euclidean distance



where,   
 - D^2 is the square of the Mahalanobis distance.   
 - x is the vector of the observation (row in a dataset),   
 - m is the vector of mean values of independent variables (mean of each column),   
 - C^(-1) is the inverse covariance matrix of independent variables.

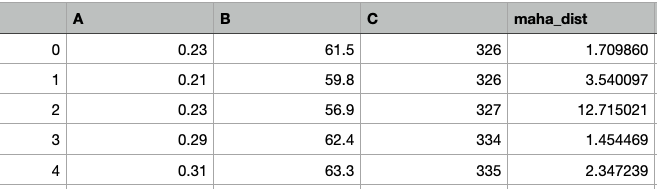
Compute Mahalanobis Distance

import pandas as pd  
import scipy as sp  
import numpy as np  
  
filepath = 'local/input[.csv](https://raw.githubusercontent.com/selva86/datasets/master/diamonds.csv)'  
df = pd.read\_csv(filepath).iloc[:, [0,4,6]]  
df.head()



Let’s write the function to calculate Mahalanobis Distance:

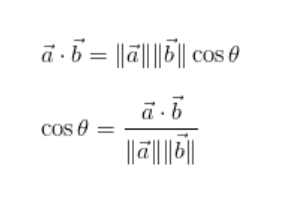
def mahalanobis(x=None, data=None, cov=None):  
 """Compute the Mahalanobis Distance between each row of x and the data   
 x : vector or matrix of data with, say, p columns.  
 data : ndarray of the distribution from which Mahalanobis distance of each observation of x is to be computed.  
 cov : covariance matrix (p x p) of the distribution. If None, will be computed from data.  
 """  
 x\_minus\_mu = x - np.mean(data)  
 if not cov:  
 cov = np.cov(data.values.T)  
 inv\_covmat = sp.linalg.inv(cov)  
 left\_term = np.dot(x\_minus\_mu, inv\_covmat)  
 mahal = np.dot(left\_term, x\_minus\_mu.T)  
 return mahal.diagonal()  
  
df\_x = df[['A', 'B', 'C']].head(500)  
df\_x['maha\_dist'] = mahalanobis(x=df\_x, data=df[['A', 'B', 'C']])  
df\_x.head()



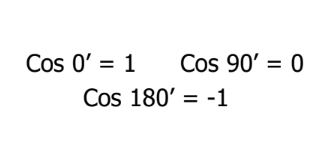
# Cosine Distance:

Mostly Cosine distance metric is used to find similarities between different documents. In cosine metrics, we measure the degree of angle between two documents/vectors(the term frequencies in different documents collected as metrics). This particular metric is used when the magnitude between vectors does not matter but the orientation.

Cosine similarity formula can be derived from the equation of dot products:-



Now, you must be thinking about which value of cosine angle will be helpful in finding out the similarities.



Now that we have the values which will be considered in order to measure the similarities, we need to know what do 1, 0, and -1 signify.

Here cosine value 1 is for vectors pointing in the same direction i.e. there are similarities between the documents/data points. At zero for orthogonal vectors i.e. Unrelated(some similarity found). Value -1 for vectors pointing in opposite directions(No similarity).

sklearn.metrics.pairwise.**cosine\_similarity**(X, Y=None, dense\_output=True)

Example for Cosine Similarity

from scipy import spatial  
  
dataSetI = [3, 45, 7, 2]  
dataSetII = [2, 54, 13, 15]  
result = 1 - spatial.distance.cosine(dataSetI, dataSetII)

Another Version based on Numpy

from numpy import dot  
from numpy.linalg import norm  
  
cos\_sim = dot(a, b)/(norm(a)\*norm(b))

Defining the Cosine Similarity function

import math  
def cosine\_similarity(v1,v2):  
 "compute cosine similarity of v1 to v2: (v1 dot v2)/{||v1||\*||v2||)"  
 sumxx, sumxy, sumyy = 0, 0, 0  
 for i in range(len(v1)):  
 x = v1[i]; y = v2[i]  
 sumxx += x\*x  
 sumyy += y\*y  
 sumxy += x\*y  
 return sumxy/math.sqrt(sumxx\*sumyy)  
  
v1,v2 = [3, 45, 7, 2], [2, 54, 13, 15]  
print(cosine\_similarity(v1,v2))

Running the example, we can see we get the same results, confirming our manual implementation.

0.972284251712

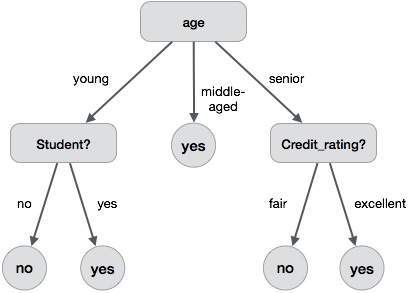
# **Decision Tree Induction**

[**Previous Page**](https://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm)

[**Next Page**](https://www.tutorialspoint.com/data_mining/dm_bayesian_classification.htm)

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy\_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

* It does not require any domain knowledge.
* It is easy to comprehend.
* The learning and classification steps of a decision tree are simple and fast.

## **Decision Tree Induction Algorithm**

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

Generating a decision tree form training tuples of data partition D

**Algorithm : Generate\_decision\_tree**

**Input:**

Data partition, D, which is a set of training tuples

and their associated class labels.

attribute\_list, the set of candidate attributes.

Attribute selection method, a procedure to determine the

splitting criterion that best partitions that the data

tuples into individual classes. This criterion includes a

splitting\_attribute and either a splitting point or splitting subset.

**Output:**

A Decision Tree

**Method**

create a node N;

if tuples in D are all of the same class, C then

return N as leaf node labeled with class C;

if attribute\_list is empty then

return N as leaf node with labeled

with majority class in D;|| majority voting

apply attribute\_selection\_method(D, attribute\_list)

to find the best splitting\_criterion;

label node N with splitting\_criterion;

if splitting\_attribute is discrete-valued and

multiway splits allowed then // no restricted to binary trees

attribute\_list = splitting attribute; // remove splitting attribute

for each outcome j of splitting criterion

// partition the tuples and grow subtrees for each partition

let Dj be the set of data tuples in D satisfying outcome j; // a partition

if Dj is empty then

attach a leaf labeled with the majority

class in D to node N;

else

attach the node returned by Generate

decision tree(Dj, attribute list) to node N;

end for

return N;

## **Tree Pruning**

Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

### Tree Pruning Approaches

There are two approaches to prune a tree −

* **Pre-pruning** − The tree is pruned by halting its construction early.
* **Post-pruning** - This approach removes a sub-tree from a fully grown tree.

## **Cost Complexity**

The cost complexity is measured by the following two parameters −

* Number of leaves in the tree, and
* Error rate of the tree.